

User Provisioning

Provisioning is the process of creating, maintaining and deleting digital identities (accounts) for users of a system(s) and linking appropriate rights to identities in the form of rules and roles. Identity provisioning is key for the Identity Federation. Identity federation is a mechanism that allows authentication across different enterprises in different trust domains based on a trust factor. This makes access easy, as users do not have to remember a different set of credentials for every application they use. See the following identity provisioning key concepts that are used in WSO2 Identity Server.

- Inbound provisioning

Inbound provisioning provisions users or groups into the WSO2 Identity Server by an external application. These external applications are referred to as service providers. WSO2 Identity Server supports the [SCIM 2.0](#) API standards for inbound provisioning.

- Outbound provisioning

Outbound provisioning provisions users to a trusted identity provider from the WSO2 Identity Server. A trusted identity provider is basically an identity provider that supports inbound provisioning. It can be Google, Salesforce, another Identity Server, etc.

This tutorial guides you to perform inbound provisioning using SCIM2, and outbound provisioning and Just-In-Time provisioning using two WSO2 Identity Servers.

Inbound provisioning Using SCIM2

Setting up:

- Download and start the WSO2 Identity Server 7.0.0

Try it :

1. Provision the user into the Identity Server

Request

```
curl -X 'POST' \
  'https://localhost:9443/scim2/Users' \
  -H 'accept: application/scim+json' \
  -H 'Content-Type: application/scim+json' \
  -H 'Authorization: Basic <base_64_encoded_username_password>' \
  \

  -d '{
    "schemas": [],
    "name": {
      "givenName": "Kim",
      "familyName": "Berry"
    },
    "emails": [
      {
        "value": "kim@wso2.com",
        "primary": true
      }
    ],
    "userName": "kim",
    "password": "MyPa33w@rd"
  }'
```

Response

```
{ "emails": ["kim@wso2.com"], "meta": { "created": "2024-04-24T14:03:28.890629Z", "location": "https://localhost:9443/scim2/Users/8ade213f-f3ad-4e49-b975-18b379ded76b", "lastModified": "2024-04-24T14:03:28.890629Z", "resourceType": "User", "schemas": ["urn:ietf:params:scim:schemas:core:2.0:User", "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User", "urn:scim:wso2:schema"], "roles": [ { "audienceValue": "10084a8d-113f-4211-a0d5-efe36b082211", "display": "everyone", "audienceType": "organization", "value": "8f38e212-cd6a-4027-9244-f851a81589e4", "$ref": "https://localhost:9443/scim2/v2/Roles/8f38e212-cd6a-4027-9244-f851a81589e4", "audienceDisplay": "Super" } ], "name": { "givenName": "Kim", "familyName": "Berry", "id": "8ade213f-f3ad-4e49-b975-18b379ded76b", "userName": "kim" }
```

2. View the provisioned users using user id

Obtain the user id from the above response.

Request

```
curl -X 'GET' \
'https://localhost:9443/scim2/Users/{user-id}' \
-H 'accept: application/scim+json'
-H 'Authorization: Basic <base_64_encoded_username_password>'
```

3. Update the provisioned user details

In response you will get the updated user profile.

Request

```
curl -X 'PUT' \
'https://localhost:9443/scim2/Users/{user-id}' \
-H 'accept: application/scim+json' \
-H 'Content-Type: application/scim+json' \
-H 'Authorization: Basic <base_64_encoded_username_password>'
-d '{
```

```

"schemas": [],
"name": {
  "givenName": "Jimmy",
  "familyName": "Kimmel"
},
"userName": "kim",
"emails": [
  {
    "value": "kim@test.com",
    "primary": true
  }
],
}'

```

Response

```

{"emails":["kim@test.com"],"meta":{"created":"2024-04-24T14:03:28.890629Z","location":"https://localhost:9443/scim2/Users/8ade213f-f3ad-4e49-b975-18b379ded76b","lastModified":"2024-04-24T14:04:41.567169Z","resourceType":"User"},"schemas":["urn:ietf:params:scim:schemas:core:2.0:User","urn:ietf:params:scim:schemas:extension:enterprise:2.0:User","urn:scim:wso2:schema"],"roles":[{"audienceValue":"10084a8d-113f-4211-a0d5-efe36b082211","display":"everyone","audienceType":"organization","value":"8f38e212-cd6a-4027-9244-f851a81589e4","$ref":"https://localhost:9443/scim2/v2/Roles/8f38e212-cd6a-4027-9244-f851a81589e4","audienceDisplay":"Super"}],"name":{"givenName":"Kim","familyName":"Berry"},"id":"8ade213f-f3ad-4e49-b975-18b379ded76b","userName":"kim"}

```

4. Delete the provisioned user by user id

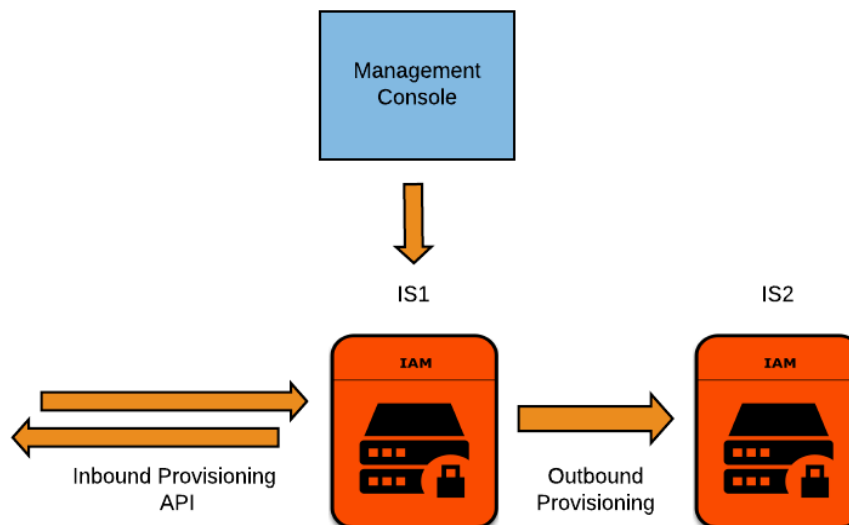
Request

```

curl -X 'DELETE' \
'https://localhost:9443/scim2/Users/{user-id}' \
-H 'accept: */*'
-H 'Authorization: Basic <base_64_encoded_username_password>'

```

Outbound Provisioning



In this section, we configure one Identity Server to provision users to another Identity Server using SCIM.

For ease, let's name the WSO2 Identity Server doing JIT provisioning as **IS1** and the identity provider WSO2 Identity Server as **IS2**.

Prerequisite: Configuring port offset on IS2

To run two IS instances at the same time, the port of one must be changed from the default value of 9443. To change the port of IS2 to 9444, do the following,

1. Navigate to `<IS2_HOME>/repository/conf` folder. Open the `deployment.toml` file in a text editor.
2. Add `offset = 1` under the `[server]` section and save the file.
3. Start IS2.

Setting up:

Step 1: Configuring an identity provider

1. Login to the WSO2 Identity Server **Console** of **IS1**, using your admin credentials (e.g. admin:admin).
2. In the WSO2 Identity Server **Console**, from the menu click **Connections**.
3. Click on **+ New Connections** and select **Custom Connector**.

Custom Connector
Create a new Connection with minimum configurations.

Name *
Enter a name for the connection.

Description
Enter a description of the connection.

Cancel Create

4. Add a unique name as the name of the connection. (for example, **SCIM-IDP**)
5. Goto the **Outbound Provisioning** tab and click **+New Connector**.
6. Select **SCIM2** as the **Connector Type**.
7. Enter the following values in the respective fields and click **Next**.

Field	Value
Username	Username of SCIM application (in this case, a username of a user registered in IS2, such as "admin")
Password	Password of the user entered in the username field
User Endpoint	https://localhost:9444/scim2/Users
Group Endpoint	https://localhost:9444/scim2/Groups
Userstore Domain	PRIMARY
Enable Password Provisioning	Keep this checked.

Create outbound provisioning connector

Follow the steps to add new outbound provisioning connector

Fill the basic information about the provisioning connector.

Connector selection Connector Details Summary

SCIM-IDP

Create a new Connection with minimum configurations.

PROVISIONING SETTINGS

Connector SCIM2

Username admin

User Endpoint https://localhost:9444/scim2/Users

Group Endpoint https://localhost:9444/scim2/Groups

User Store Domain PRIMARY

Enable Password Provisioning true

Cancel Previous Finish

8. Click **Finish** to save your changes.
9. Now you will see the **SCIM2** listed under the **Outbound Provisioning Connectors**.

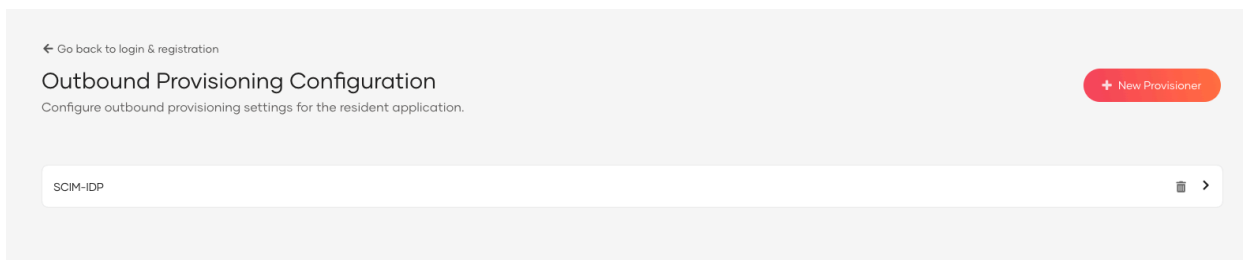
The screenshot displays the WSO2 Identity Server Console interface. On the left is a sidebar menu with options: Home, Applications, Connections (highlighted), API Resources, Branding, User Management, User Attributes & Stores, Organizations, Login & Registration, Email & SMS, Console Settings, and Server. The main content area shows the 'SCIM-IDP' connector configuration page. At the top, there's a 'Go back to Connections' link and a 'Custom Connector' button with the text 'Create a new Connection with minimum configurations.' Below this is a tabbed interface with tabs: General, Settings, Attributes, Connected Apps, Groups, Outbound Provisioning (active), Just-in-Time Provisioning, and Advanced. The 'Outbound Provisioning' tab contains a section titled 'Outbound Provisioning Connectors' with a '+ New Connector' button. Below this is a list of connectors, currently showing 'SCIM2' with a 'Disabled' toggle switch. Underneath is a section for 'OutBound Provisioning Roles' with a 'Role' dropdown menu set to 'Select Role' and a '+ Add' button. A note states: 'Select and add as identity provider outbound provisioning roles'. An 'Update' button is at the bottom of the roles section.

10. Click on the **Toggle Button** to enable the Connector.

Step 2: Configuring the resident service provider

When configuring outbound provisioning for any user management operation done via the console, outbound provisioning identity providers must be configured against the resident service provider. So, based on the outbound configuration, users added from the console can also be provisioned to external systems, in this case, IS2.

1. Login to the WSO2 Identity Server **Console** of **IS1**, using your admin credentials (e.g. admin:admin).
2. In the WSO2 Identity Server **Console**, from the menu click **Login & Registration**.
3. Go to **Provisioning Settings > Outbound Provisioning Configuration** and click **New Provisioner**.
4. Select the connection in which you have configured outbound provisioning as the **Connection. (SCIM-IDP)**
5. Select the relevant outbound connector as the **Provisioning Connector. (SCIM2)**
6. Click **Finish**.



← Go back to login & registration

Outbound Provisioning Configuration

Configure outbound provisioning settings for the resident application.

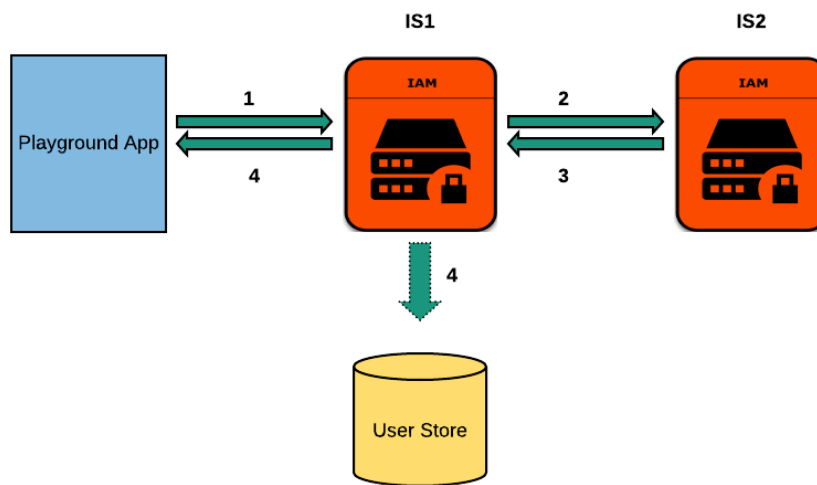
New Provisioner

SCIM-IDP

Try it

1. Log in to the **Console** of **IS1**.
2. Add a new user.
 - a. Expand the **User Management** menu item, click on **Users**.
 - b. On the **Users** page, click **Add User** and select **Single User** from the dropdown.
 - c. Fill all required fields and click **Finish**.
3. Log in to the management console of IS2.
4. Expand the **User Management** menu item, click on **Users**.
5. You will observe that the user added to IS1 has been added to the IS2 userstore as well.

Just-In-Time Provisioning



JIT provisioning is a means of provisioning users into the Identity Server at the time of federated authentication. A service provider initiates the authentication request, the user gets redirected to the Identity Server, and then the Identity Server redirects the user to an external identity provider for authentication. JIT provisioning gets triggered only when the Identity Server receives a positive authentication response from the external identity provider. The Identity Server will provision the user to its internal user store with the user claims from the authentication response.

In this section, we discuss how to set up one Identity Server for JIT provisioning from another Identity Server acting as an identity provider.

For ease, let's name the WSO2 Identity Server doing JIT provisioning as **IS1** and the identity provider WSO2 Identity Server as **IS2**.

Setting Up

Step 1: Configure a service provider on IS2

1. Start IS2.
2. Login to the WS02 Identity Server **Console** and from the menu click **Applications**.
3. Click **New Application**.
4. From the given set of templates select **Traditional Web Application** template.
5. Add name as SP-IS2 and Authorized redirect URLs as <https://localhost:9443/commonauth>.
6. Click **Create**.
7. Make a note of the **Client ID** and **Client Secret** of this Service Provider.

Step 2: Configuring IS2 as an identity provider with JIT provisioning in IS1

1. Start IS1.
2. Login to the WS02 Identity Server **Console**, from the menu click **Connections**.
3. Click + **New Connection**.
4. From the available templates select **Standard based Connections** template.
5. Add **IDP-IS2** as the Identity provider name and click **Next**.
6. Expand the **Federated Authenticators** section, expand **OAuth/OpenID Connect Configuration** subsection and fill the following fields

Field	Value
Client ID	Enter the client ID of the service provider registered in IS2
Client Secret	Enter the client secret of the service provider registered in IS2
Authorization Endpoint URL	https://localhost:9444/oauth2/authorize
Token Endpoint URL	https://localhost:9444/oauth2/token

- Goto the Just-In-Time-Provisioning tab.
- Check** Just-In-Time-Provisioning option to enable JIT provisioning.

The screenshot shows the WSO2 Identity Server Console interface. On the left is a navigation menu with options: Home, Applications, Connections (selected), API Resources, Branding, User Management (Users, Groups, Roles), User Attributes & Stores, Organizations, Login & Registration, Email & SMS, Console Settings, and Server. The main content area is titled 'IDP-IS2' and includes a 'Go back to Connections' link. Below the title are tabs: General, Settings, Attributes, Connected Apps, Groups, Outbound Provisioning, Just-in-Time Provisioning (active), and Advanced. The 'Just-in-Time (JIT) User Provisioning' section is expanded, showing a checkbox that is checked. Below this, a dropdown menu is set to 'PRIMARY'. The 'Provisioning scheme' section has four radio button options: 'Prompt for username, password and consent', 'Prompt for password and consent', 'Prompt for consent', and 'Provision silently' (which is selected). An 'Update' button is at the bottom of the form.

9. Select **PRIMARY** in the dropdown to provision users to the primary userstore, and then select **Provision silently** to allow the provisioning process to complete without prompting for extra details (for demonstration purposes).
10. Click **Update**.

Step 3: Configuring a service provider in IS1 to federate authentication to IS2

We will be using the [playground2](#) sample app to test the flow. Therefore, as a prerequisite, the playground2 app must be deployed in Tomcat.

1. In the WSO2 Identity Server **Console**, from the menu click **Applications**.
2. Click **New Application**.
3. From the given set of templates select **Traditional Web Application** template.
4. Fill the fields in the create application wizard.

Name: *playground2*

Protocol: *OpenID Connect*

Authorized redirect URLs:

<http://localhost:8080/playground2/oauth2client>

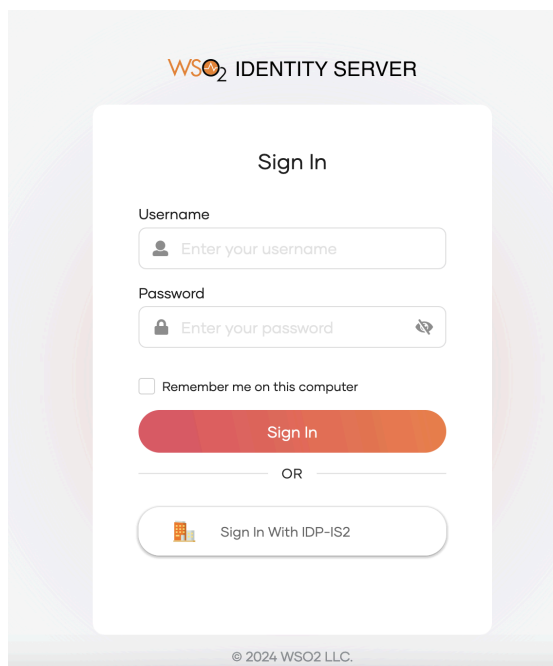
5. Make a note of the **Client ID** of this registered OAuth/OpenID Connect service provider.
6. Click on the **Login Flow tab**.
7. Add a new sign in option to the first step. Click on **+Add Sign In Option Button**.
8. Select **IDP-IS2** and click **Update**.

Step 4: Add a user in IS2

1. Log in to the management console of IS2.
2. Add a new user.
 - a. Expand the **User Management** menu item, click on **Users**.
 - b. On the **Users page**, click **Add User** and select **Single User** from the dropdown.
 - c. Fill all required fields and click **Finish**.

Try it

1. Access the playground2 app via <http://localhost:8080/playground2/oauth2.jsp>
2. Enter the client ID of the service provider registered in IS1 in the **Client ID** field.
3. Click **Authorize**. The browser will be redirected to the login page of IS1.
4. From that login page select the IDP-IS2 option to authenticate users from the IS2.

The image shows the WSO2 Identity Server Sign In page. At the top, it says "WSO2 IDENTITY SERVER". Below that is a "Sign In" heading. There are two input fields: "Username" with a placeholder "Enter your username" and "Password" with a placeholder "Enter your password" and a toggle icon. Below the password field is a checkbox labeled "Remember me on this computer". There is a large orange "Sign In" button. Below the button is an "OR" separator. Below the separator is a button with a user icon and the text "Sign In With IDP-IS2". At the bottom, it says "© 2024 WSO2 LLC."

5. Enter the credentials of the user registered in IS2 and click **Login**.
6. Once the authentication is successful, log in to the management console of IS1 and list the users. You will observe that the user registered in IS2 has now been added to the userstore of IS1 as well.